



Janne Hannila

MIKROPROSESSORIOHJATTU JÄRJESTELMÄ

MIKROPROSESSORIOHJATTU JÄRJESTELMÄ

Janne Hannila
Opinnäytetyö
Kevät 2012
Tietotekniikan koulutusohjelma
Oulun seudun ammattikorkeakoulu

TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu
Tietotekniikan koulutusohjelma, elektroniikan suunnittelu ja testaus

Tekijä: Janne Hannila
Opinnäytetyön nimi: Mikroprosessoriohjattu järjestelmä
Työn ohjaaja: Timo Vainio
Työn valmistumislukukausi ja -vuosi: Kevät 2012
Sivumäärä: 28 + 3 liitettä

Opinnäytetyön aiheena oli tehdä Hannila Groupille mikroprosessoriohjattu järjestelmä, jolla voidaan ohjata järjestelmään kytkettyjä LED-valoja. Tavoitteena oli toteuttaa toimiva järjestelmä, jossa hyödynnetään Arduino-kortteja alustana. Lisäksi tavoitteena oli suunnitella piirikortti, johon tulisi LED-valojen himmentämiseen tarvittavat komponentit, sekä tuottaa ohjelmakoodi, joka toteuttaa halutun toiminnan.

Työ onnistui hyvin ja järjestelmä toimi odotetulla tavalla. Työ tuli valmiiksi suunnitelman mukaisesti ja työtä on tarkoitus jatkokehittää sekä lisätä järjestelmään toiminnallisuuksia.

Asiasanat:
Arduino, mikrokontrolleri, PWM, LED, himmennin, I2C

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Information Technology, Electronics Design and Testing

Author: Janne Hannila

Title of thesis: Microprocessor Controlled System

Supervisor: Timo Vainio

Term and year when the thesis was submitted: Spring 2012

Pages: 28 + 3 appendices

Idea of my thesis was to design and implement a microprocessor controlled system for Hannila Group. This system can be used to control the lights, which are connected to it. Also the aim was to make an implementation by using the Arduino boards as platform. Also, the aim was to design circuit board, having several components for dimming LED-lights and to produce the program code carrying out a desired functionality.

The thesis project was successful and the designed system worked as expected. The empirical part of thesis work was completed as planned. There are some possibilities to further develop and increase the system functionality.

Keywords:

Arduino, microcontroller, PWM, LED, dimmer, I2C

ALKULAUSE

Haluan kiittää Hannila Groupia työn toimeksiannosta sekä Esa Hannilaa työn johtamisesta. Lisäksi haluan kiittää Oulun seudun ammattikorkeakoulun opettajia Timo Vainiota työn ohjaamisesta sekä Tuula Hopeavuorta opinnäytetyön tekstin ohjaamisesta.

Oulussa 3.6.2012

Janne Hannila

SISÄLLYS

TIIVISTELMÄ	3
ABSTRACT	4
ALKULAUSE	5
SISÄLLYS	6
SANASTO	8
1 JOHDANTO	9
2 PULSSINLEVEYSMODULAATIO	10
2.1 Analogiaelektroniikka	10
2.2 Digitaalinen ohjaus	11
2.3 Logic-level-transistori	13
3 I2C-VÄYLÄ	14
3.1 Käyttökohteet	14
3.2 Tekniikka	14
3.2.1 Isäntä ja orja	14
3.2.2 Moni-isäntäinen väylä	15
3.2.3 Data ja kello	15
3.2.4 Nopeus	16
3.2.5 Tiedonsiirto	16
4 ARDUINO-KORTTI	18
4.1 Käyttö	18
4.2 Arduino Unon tekniset tiedot	18
5 LAITTEISTON TOTEUTUS	20
5.1 Lohkotason ratkaisu	20
5.2 Piirilevyn suunnittelu	20
5.3 Toiminnankuvaus	21
6 OHJELMISTON TOTEUTUS	22
6.1 Lohkotason ratkaisu	22
6.2 Ohjelmanrakenne	23
6.3 Toiminnankuvaus	23
7 TESTAUS	25

8 YHTEENVETO	26
LÄHTEET	27
LIITTEET	28

SANASTO

BJT	Bipolar Junction Transistor, bipolaaritransistori
DC	Direct Current, tasavirta
DesignSpark PCB	Vapaasti ladattavissa piirilevyn suunnitteluohjelma
DSP	Digital Signal Processor, digitaalinen signaaliprosessori
DuinOS	Reaaliaikakäyttöjärjestelmä
FreeRTOS	Reaaliaikakäyttöjärjestelmä
Kernel	Käyttöjärjestelmän ydin
MOSFET	Metal-oxide-semiconductor field-effect transistor, metalli-oksidi-puolijohdekanavatransistori
Open collector	Avolähtö, jossa BJT-transistorin kollektori toimii IC-piirin lähtönä
Open drain	Avolähtö, jossa FET-transistorin nielu toimii IC-piirin lähtönä
OrCAD	Cadencen valmistama piirilevysuunnitteluohjelma.
Processing	Avoimen lähdekoodin ohjelmointikieli ja -ympäristö
PWM	Pulse Width Modulation, pulssinleveysmodulaatio
RS232	Sarjaväyläinen tietoliikenneportti
Task	Taski, moniajossa rinnakkain suoritettava tehtävä
Timer	Mikroprosessorin ajastin
Wiring	Avoimen lähdekoodin ohjelmointikehys mikrokontrollereille

1 JOHDANTO

Mikroprosessoriohjattu järjestelmä on laitteistokokonaisuus. Se muodostuu useasta piirikortista, joissa on mikrokontrolleri sekä komponentteja. Tässä työssä käytettiin kaupallista Arduino Uno -korttia järjestelmän pohjana sekä lisäksi suunniteltiin oma piirikortti, johon tuli tarvittavat ulkoiset komponentit himmentyksen toteuttamiseksi. Tähän ratkaisuun päädyttiin, jotta työhön käytettävä aika riittäisi toimivan ohjelmakoodin tuottamiseen eikä menisi oman mikroprosessorikortin suunnitteluun.

Työn tavoitteena oli suunnitella ja toteuttaa järjestelmä, joka kykenee suorittamaan valaistuksenohjauksen LED-valaisimille moniajona, ilman että käyttäjän tarvitsee ymmärtää järjestelmän toimintaa syvällisemmin. Järjestelmään voidaan kytkeä useita valokatkaisijoita, joilla valoja ohjataan, sekä useita LED-valoja, joita halutaan himmentää. Moniajon lisääminen järjestelmään vaati reaaliaikakäyttöjärjestelmän hyödyntämistä ohjelmakoodissa.

Liitteessä 1 on opinnäytetyön aloituspäätöksen tehty lähtötietomuuisto. Osa tämän työn liitteistä on liikesalaisuuksina saatavissa vain yrityksen sisäiseen käyttöön, joten niitä ei tähän dokumenttiin sisällytetä.

2 PULSSINLEVEYSMODULAATIO

PWM eli pulssinleveysmodulaatio on tehokas ohjaustekniikka analogiapiireissä. PWM-signaali voidaan muodostaa prosessorin digitaallilähdöillä. PWM:ää käytetään laajasti monissa erilaisissa sovelluksissa mittauksesta ja viestinnästä tehonsäätön ja muuntamiseen. (1.)

2.1 Analogiaelektroniikka

Analogisella signaalilla on jatkuvasti muuttuva arvo äärettömällä resoluutiolla sekä aika- että voimakkuustasossa. 9 voltin paristo on esimerkki analogisesta laitteesta, sillä sen ulostulojännite ei ole tarkasti 9 V, vaan se vaihtelee ajan kanssa ja voi saada minkä tahansa reaalisen arvon. Samoin tavoin paristosta otetun virran suuruus ei ole rajattu tiettyyn määrään mahdollisia arvoja. Digitaalinen signaali on erotettavissa analogisesta signaalista, koska digitaalinen signaali voi saada vain ennalta määrättyjä arvoja, kuten 0 V ja 5 V. (1.)

Analogisilla jännitteillä ja virroilla voidaan suoraan ohjata asioita, kuten autoradion äänenvoimakkuutta. Yksinkertaisessa analogisessa radiossa äänenvoimakkuudensäädin onkin säätövastus. Kun nuppia käännetään, niin vastus muuttuu pienemmäksi tai isommaksi. Tällöin vastuksen läpi kulkeva virta kasvaa tai pienenee. Tämä muuttaa kaiuttimelle ohjattavaa virtaa suurentaen tai pienentäen äänenvoimakkuutta. Analogisen piirin, kuten radion, lähtö on lineaarisessa suhteessa sen tuloon. (1.)

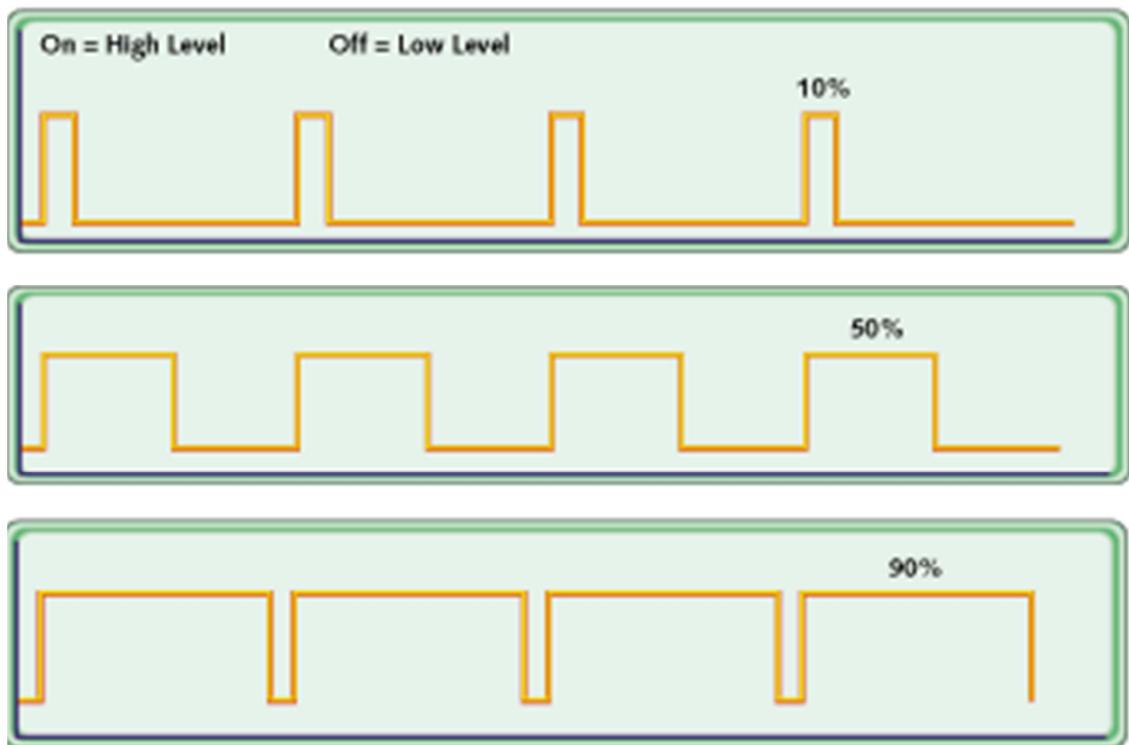
Vaikka analoginen ohjaus tuntuu yksinkertaiselta, se ei kuitenkaan ole välttämättä ekonomisesti tai muuten kiinnostava. Yhtenä asiana on se, että analogisilla piireillä on taipumusta muuttua ajan saatossa ja niitä voi täten olla todella hankala virittää. Tarkat analogiset piirit, jotka ratkaisevat tämän ongelman, voivat olla todella suuria, painavia ja kalliita. Analogiset piirit voivat myös kuumeta. Tehonkulutus on suhteessa aktiivisten elementtien yli oleva jännite kerrottuna läpi kulkevalla virralla. Analogiset piirit ovat usein myös hyvin herkkiä kohinalle. Äärettömän resoluution takia mikä tahansa häiriö tai kohina analogisessa signaalissa muuttaa virran arvoa. (1.)

2.2 Digitaalinen ohjaus

Ohjaamalla analogisia piirejä digitaalisesti voivat järjestelmän kustannukset ja tehonkulutus olla rajusti pienemmät. Lisäksi monet mikrokontrollerit ja DSP:t sisältävät sisäisen PWM-ohjaimen, joka tekee käyttöönoton helpoksi. (1.)

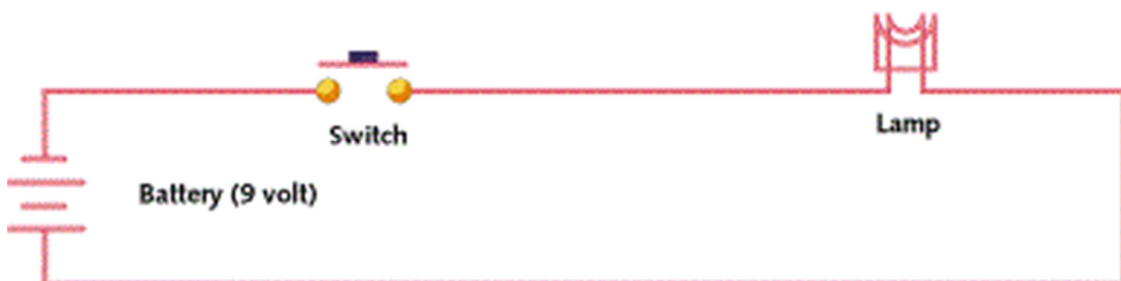
Lyhykäisyydessään PWM on keino koodata digitaalisesti analogisia signaali-tasoja. Korkearesoluutioisia laskureita käyttämällä kanttiaallon pulssisuhde on moduloitu, että saadaan koodattua tietty analoginen signaalitaso. PWM-signaali on silti digitaalinen signaali, koska millä tahansa ajanhetkellä digitaalinen lähtö on joko loogisessa ykkösessä (esim. 5 V) tai nollassa (0 V). Jännite- tai virtalähteeltä syötetään haluttu taso analogiselle kuormalle toistuvien päälle-pois-pulssisarjojen avulla. Päällä olon aikana DC-lähde on liitettynä kuormaan ja pois päältä -jakson aikana lähde on kytketty pois päältä. Riittävällä kaistanleveydellä mikä tahansa analoginen arvo voidaan koodata PWM:llä. (1.)

Kuva 1 näyttää kolme erilaista PWM-signaalia. Kuvassa 1a on PWM-lähtö 10 %:n pulssisuhteella. Tästä kuvasta näkee, että signaali on päällä 10 %:n jakson ja pois päältä loput 90 %. Kuvissa 1b ja 1c ovat vastaavasti PWM-lähdöt 50 %:n ja 90 %:n pulssisuhteella. Nämä kolme PWM-lähtöä koodaavat kolme erilaista analogista arvoa, tasoilla 10 %, 50 % ja 90 % täydestä voimakkuudesta. Jos esimerkiksi lähdejännite on 9 voltia ja pulssisuhde 10 %, saadaan tuloksena 0,9 voltin analoginen signaali. (1.)



KUVA 1. PWM-signaaleja eri pulssisuhteilla (1)

Kuvassa 2 on yksinkertainen piiri, jota voidaan ohjata PWM:llä. Kuvassa oleva 9 voltin paristo syöttää hehkulamppua. Jos kytkin suljetaan 50 millisekunniksi, yhdistyy paristo lamppuun ja se saa 9 voltia. Jos kytkin avataan seuraavan 50 millisekunnin ajaksi, lamppu saa 0 voltia. Jos tätä sykliä toistetaan 10 kertaa sekunnissa, lamppu palaa kuten se olisi kytkettynä 4,5 voltin paristoon (50 % 9 voltista). Näin voidaan todeta, että pulssisuhte on 50 % ja moduloiva taajuus on 10 hertsiä. (1.)



KUVA 2. Yksinkertainen PWM-piiri (1)

Suurin osa kuormista, induktiiviset ja kapasitiiviset, tarvitsee paljon korkeamman modulaatiotaajuuden kuin 10 hertsiä. Esimerkiksi kun lamppu kytketään

päälle 5 sekunniksi, sen jälkeen pois päältä 5 sekunniksi ja päälle uudestaan, pulssisuhde olisi 50 %, mutta polttimo palaisi kirkkaasti ensimmäiset 5 sekuntia, jonka jälkeen toiset 5 sekuntia se olisi pois päältä. Jotta polttimo saisi 4,5 voltin jännitteen, syklin jaksonaika täytyy olla lyhyt suhteessa kuorman vasteaikaan. Jotta saavutettaisiin haluttu himmennysefekti lampulle, on tarpeellista kasvattaa modulointitaajuutta. Sama pätee myös muihin PWM-sovelluksiin. Yleinen modulaatiotaajuuksien alue on 1 kHz:stä 200 kHz:iin. (1.)

2.3 Logic-level-transistori

Logic-level- eli logiikkatason transistori on komponentti, joka soveltuu erittäin hyvin mikrokontrollerin digitaalisella lähdöllä ohjattavaksi. Komponentin nielun ja lähteen väli johtaa täysin jo 5 voltin hilajännitteellä. Se myös soveltuu hyvin PWM-ohjaukseen ilman erillisiä komponentteja. Normaalit MOSFET-transistorit eivät yleensä sovellu kyseiseen tarkoitukseen niiden suuren hilajännitteen vuoksi.

3 I2C-VÄYLÄ

3.1 Käyttökohteet

I2C-väylän on suunnitellut Philips 80-luvun alusta mahdollistamaan helpon kommunikoinnin samalla piirilevyllä sijaitsevien komponenttien välillä. Philips Semiconductors muutti nimensä NXP:ksi vuonna 2006. Nimi I2C kääntyy sanaksi "Inter IC". Väylää kutsutaan usein myös IIC- tai I²C-väyläksi. (2.)

Alun perin määritetty yhteysnopeus oli maksimissaan 100 kbit/s, eikä moni sovellus tarvinnut nopeampaa tiedonsiirtoa. Niille, jotka tarvitsivat, oli vaihtoehtona 400 kbit/s:n "fastmode". Myöhemmin vuonna 1998 saataville tuli "high speed" 3,4 Mbit/s:n nopeus. Viimeisimpänä määriteltiin "fast mode plus" -siirtonopeus kahden edellisen väliin. Tämä tarkoitti, että unohdettiin 400 kbit/s:ssa nopeusrajoitus ja käytettiin suurempaa nopeutta fyysisen väylän rajojen mukaan. (2.)

I2C:tä ei ole käytetty pelkästään samalla piirilevyllä olevien komponenttien yhdistämiseen, vaan myös yhdistämään komponentteja, jotka on yhdistetty kaapelilla toisiinsa. Yksinkertaisuus ja joustavuus ovat tärkeimpiä ominaisuuksia, jotka tekevät tästä väylästä kiinnostavan monille sovelluksille. (2.)

Merkittävimpiä ominaisuuksia ovat seuraavat: Tarvitaan vain kaksi väylälinjaa. Ei ole tiukkoja rajoituksia tiedonsiirtonopeuksissa kuten RS232:ssa, vaan isäntä muodostaa kellopulssin ja määrittää yhteysnopeuden. Yksinkertainen isäntä-orjasuhde sisältyy kaikkiin komponentteihin. Jokaisella laitteella, joka yhdistyy väylään, on ohjelmistolla osoitettava yksilöllinen osoite. I2C on todellinen moni-isäntäinen väylä, joka sisältää sovittelun ja törmäyksen tunnistamisen. (2.)

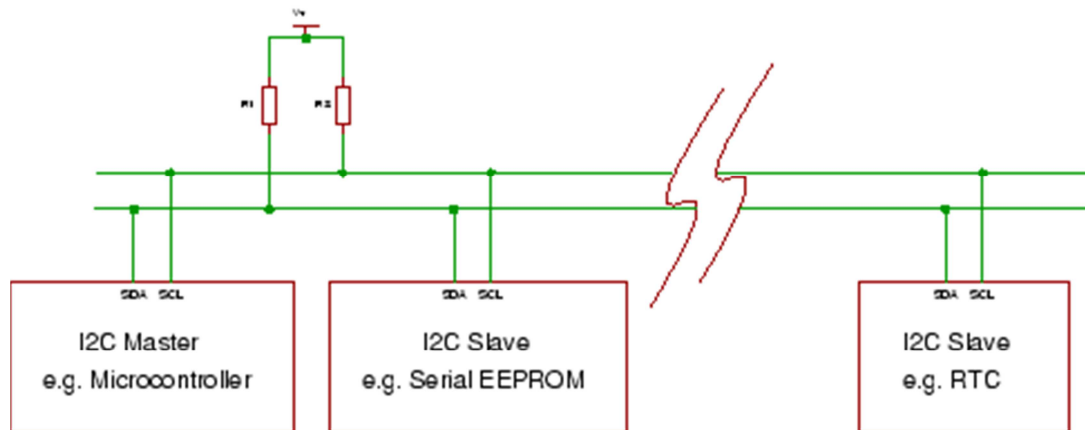
3.2 Tekniikka

3.2.1 Isäntä ja orja

Philipsin I2C-protokolla määrittää isäntä- ja orjalaitteet. Isäntälaitte hallitsee väylää sekä ohjaa kelloa ja muodostaa START- ja STOP-signaalit. Orja kuuntelee

väylää ja käyttäytyy ohjauksen sekä lähetetyn datan mukaisesti. Isäntä voi lähettää tai vastaanottaa dataa orjalle. Orjat eivät voi keskenään siirtää dataa. (4.)

Kuvasta 3 nähdään I2C-väylän rakenne. Siitä ilmenee, että väylän molemmilla linjoilla on ylösvetovastukset.



KUVA 3. I2C-väylän lohkokaavio (4)

3.2.2 Moni-isäntäinen väylä

I2C-väylällä voidaan käyttää useampaa yhtäaikaista isäntää, mutta niiden käyttö on monimutkaisempaa. Se sallii useita ohjaavia laitteita samalla väylällä. Tätä tarvitsee käyttää vain silloin, jos väylään kytketään useampi mikrokontrolleri ja jos useamman kuin yhden niistä täytyy toimia väylän isäntänä. (4.)

Moni-isäntäiseen käyttöön sisältyy väylänsovittelu, jossa isäntien täytyy kilpailla päästäkseen ohjaamaan väylää ja saadakseen kellontahdistuksen. Tässä työssä ei tutustuta tarkemmin moni-isäntäisyyteen. (4.)

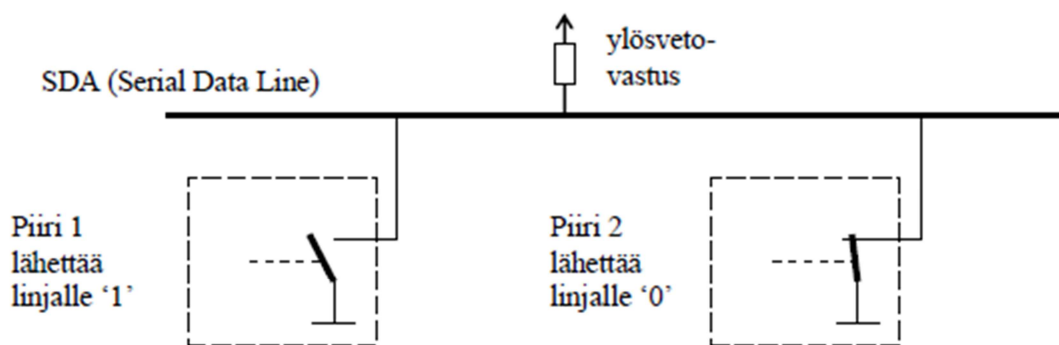
3.2.3 Data ja kello

I2C-väylä käyttää kahta kaksisuuntaista linjaa. Linjat on kello (SCL) ja data (SDA). Väylä vaatii toimiakseen myös yhteisen maapotentiaalin. Vaikka isäntälaitte ohjaa kelloa suurimman osan ajasta, voivat orjat kuitenkin vaikuttaa kellosignaaliin käskien isäntää pysähtymään hetkeksi. (4.)

Hitaat orjalaitteet saattavat joutua pysäyttämään väylän liikenteen datankeräyksen ajaksi tai keskeytyksen vuoksi. Orja voi tehdä sen pitämällä kellolinjan

(SCL) loogisessa nollassa pakottaen isännän menemään odotustilaan. Ennen jatkamista isännän täytyy odottaa, kunnes kello on vapautettu. (4.)

Väylän molempia linjoja täytyy ohjata open collector, tai open drain -lähdeillä sekä molemmilla linjoilla täytyy olla ylösvetovastus. Minkä tahansa laitteen vetäessä linjan loogiseen nollatilaan näkevät myös muut linjaan kytketyt laitteet loogisen nolla-arvon. Loogiseen ykköstilaa saavuttamiseksi kaikkien laitteiden täytyy lopettaa linjanohjaaminen eli avata kytkin, kuten kuvasta 4 näkyy. Avolähdössä transistori toimii kytkimenä. (4.)



KUVA 4. I2C-väylän avolähdön toimintaperiaate (5)

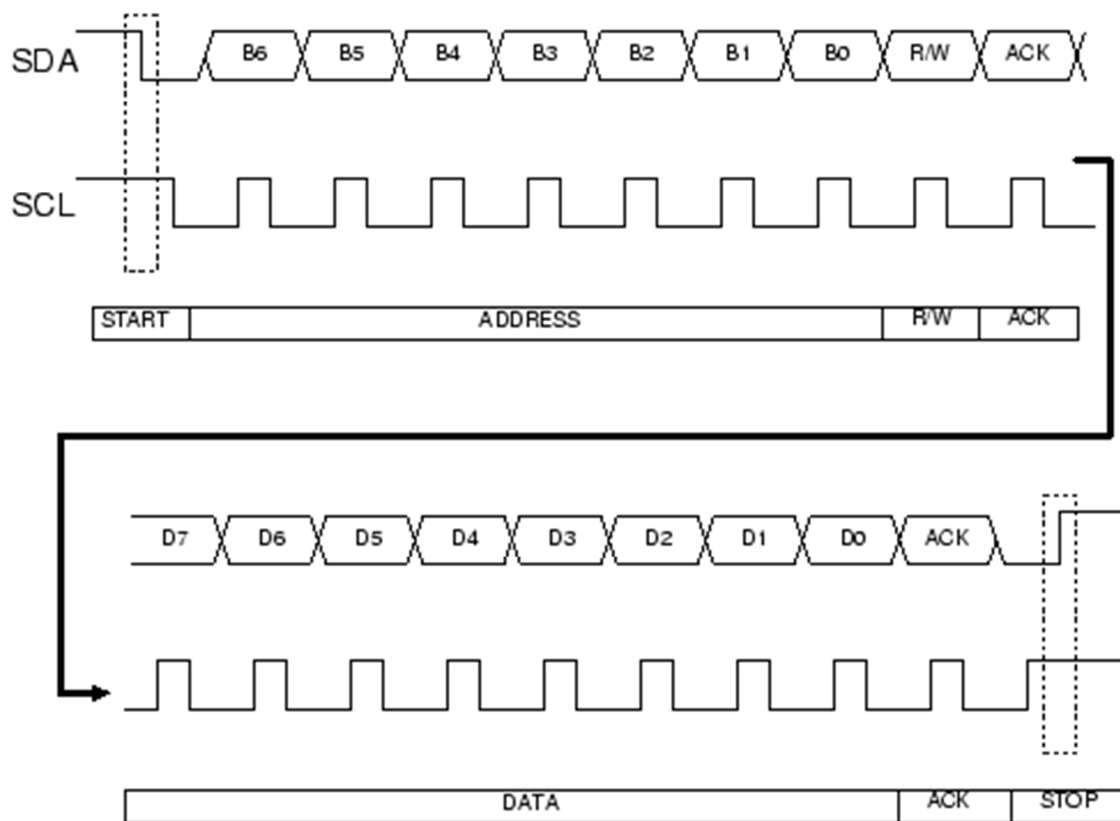
Joissakin tapauksissa laitteen valmistaja ilmoittaa "kaksijohtoinen liitäntä", vaikka se olisi I2C-yhteensopiva. Valmistaja välttää rojaltien maksamisen jättämällä I2C-sanan pois laitteen tiedoista. (4.)

3.2.4 Nopeus

Väylän normaaleja kellotaajuuksia on 10 kHz ja 100 kHz, mutta standardi sallii käyttää kellotaajuuksia nollan ja 100 kHz:n väliltä mielen mukaan. Saatavilla on myös 400 kHz:n ja 3,4 MHz:n kellotaajuudet. Standardi sallii myös käyttää vapaasti kellotaajuuksia 400 kHz:n ja 3,4 MHz:n väliltä. (4.)

3.2.5 Tiedonsiirto

Normaali isännältä orjalle luku tai kirjoitus I2C-väylällä ilmenee kuvasta 5.



KUVA 5. Tyypilliset SDA- ja SCL-signaalit I2C-väylässä (4)

Väylälle lähetetään aluksi START-bitti, eli molemmat linjat ovat aluksi 1-tilassa ja data-linja lasketaan ensin 0-tilaan, jonka jälkeen vasta kellolinja. Tämän jälkeen isäntä lähettää väylälle 8 bittiä, josta 7 ensimmäistä bittiä on orjan osoite ja viimeinen bitti kertoo, haluaako isäntä kirjoittaa vai lukea. Seuraavaksi isäntä odottaa kuittausta (ACK) orjalta, jonka osoite lähetettiin väylälle. Sitten lähetetään tai vastaanotetaan 8 bittiä dataa ja odotetaan tai lähetetään kuittaus. Loppuun tehdään kuvan 5 mukainen STOP-bitti, jolla väylä vapautetaan. (4.)

4 ARDUINO-KORTTI

Arduino on avoimen lähdekoodin elektroniikkaprototyyppialusta, joka pohjautuu joustavaan, helppokäyttöiseen laitteistoon ja ohjelmistoon. Se on tarkoitettu artisteille, suunnittelijoille, harrastelijoille ja kelle tahansa, kuka on kiinnostunut luomaan interaktiivisia objekteja tai ympäristöjä. (3.)

4.1 Käyttö

Arduino voi tunnistaa ympäristöä vastaanottamalla sisääntuloja eri antureilta ja voi vaikuttaa ympäristöönsä ohjaamalla valoja, moottoreita sekä muita toimilaitteita. Piirillä oleva mikrokontrolleri ohjelmoidaan käyttämällä Arduino-ohjelmointikieltä (pohjautuu Wiringiin) ja Arduino-kehitysympäristöä (pohjautuu Processingiin). Arduino-projektit voivat olla erillisiä tai ne voivat kommunikoida tietokoneella pyörivän ohjelmiston kanssa, kuten esimerkiksi Flash, Processing, MaxMSP. (3.)

Piirilevyt voidaan rakentaa käsin tai ostaa esikasattuna. Ohjelmiston voi ladata vapaasti valmistajan sivuilta. Laitteistojen suunnitelmat (CAD tiedostot) ovat saatavilla avoimen lähdekoodin lisenssin alaisena, ja niitä voi vapaasti hyödyntää omiin tarpeisiin. (3.)

Arduino-alustoja on useita malleja, kuten Uno, Leonardo, Mega 2560, LilyPad, Mega ADK, Fio, Ethernet, Pro, BT, Nano, Mini ja Pro Mini (3). Alustat eroavat toisistaan muistien osalta sekä myös digitaalisten ja analogisten pinnien määrässä. Lisäksi alustojen fyysiset koot eroavat toisistaan. Useat Arduino-alustat on suunniteltu siten, että niihin voidaan liittää erilaisia shieldejä, kuten Bluetooth, Ethernet, ZigBee ja GPS. Shieldit ovat eräänlaisia lisäkortteja, jotka sopivat suoraan Arduino-kortin päälle.

4.2 Arduino Unon tekniset tiedot

Arduino Uno -kortti käyttää Atmelin ATmega328 -mikrokontrolleria, joka toimii 5 voltin jännitteellä. Kortilla on sisäänrakennettu regulaattori, jonka vuoksi sille voi syöttää jännitettä 6–20 voltia, mutta suositeltava jännite on 7–12 voltia, ettei

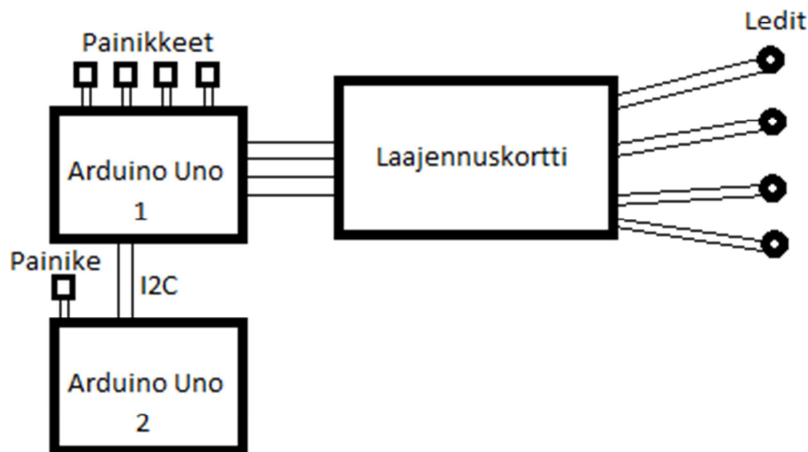
regulaattori kuumene liikaa. Kortilla on 14 digitaalista tulo-lähtöpinniä, joista kuusi voi toimia PWM-lähtönä, sekä kuusi analogista tulopinniä. Virransyöttö tulo-lähtöpinniä kohden on 40 mA ja 3,3 voltin pinnille 50 mA. Kortilla on 32 kt flash-muistia, josta käynnistyslatain käyttää 0,5 kt, 2 kt SRAM-muistia, 1 kt EEPROM-muistia ja mikrokontrollerin kellotaajuus on 16 MHz. (7.)

5 LAITTEISTON TOTEUTUS

Opinnäytetyön aiheena oli järjestelmä, jolla voidaan mikroprosessoriohjatusti hallita valaistusta ja sen himmennystä. Järjestelmä on suunniteltu talojen valaistuksen ohjaukseen. Järjestelmä ei nykyisellä kokoonpanolla riitä omakotitalon valaistuksen ohjaukseen, mutta on kuitenkin laajennettavissa tarpeen mukaan.

5.1 Lohkotason ratkaisu

Järjestelmän pohjana työssä toimii kaksi Arduino Uno -korttia, jotka kytkeytyvät toisiinsa I2C-väylän välityksellä. Järjestelmään suunniteltiin myös laajennuskortti, joka oli kytkettynä toiseen Arduino-kortista. Apuna käytettiin myös koekyt-kentälevyä johdotusten helpottamiseksi. Kuvassa 6 on järjestelmän lohkokaa-vio. Arduino 1 -korttiin oli kytkettynä neljä painonappia sekä laajennuskortti. Ar-duino 2 -korttiin kytkettiin yksi kytkin. Laajennuskorttiin kytkettiin neljä LED-valoa.



KUVA 6. Järjestelmän lohkokaa-vio

5.2 Piirilevyn suunnittelu

Järjestelmän toteuttamista varten piti suunnitella oma laajennuskortti, jonka avulla järjestelmällä voitaisiin ohjata LED-valoja. Laajennuskortin suunnitelmat

löytyvät liitteestä 2, mutta ne ovat liikesalaisuuksina saatavissa vain yrityksen sisäiseen käyttöön. Kortti suunniteltiin DesignSpark PCB -ohjelmistolla, joka on ladattavissa vapaasti valmistajan sivustosta. Tähän ohjelmistoon päädyttiin, koska sillä piirikortin suunnittelu oli helppoa sekä piirilevyn jyrsiminen onnistui koulun jyrsimellä helposti, koska ohjelmasta sai kuvat suoraan jyrsimelle sopivassa muodossa. Yksi syy oli myös se, että ohjelmasta ei syntynyt kustannuksia ja piirilevyn suunnittelu on sillä huomattavasti nopeampaa kuin esimerkiksi OrCAD-ohjelmalla.

Piirilevymateriaalina oli epoksihartsilla kyllästetty lasikuitulevy molemminpuolisella kuparipinnoitteella, johon jyrsimellä kaiverrettiin johdotukset sekä läpiviennit. Jyrsimisen jälkeen piirikortille ladottiin komponentit paikalleen ja juotettiin käsin tinalla ja kolvilla.

5.3 Toiminnankuvaus

Arduino Uno -kortteihin suunniteltiin ohjelmakoodit, joiden avulla järjestelmä toimii. Kortit toimivat järjestelmän pohjana. Korttiin numero 1 liitettiin neljä painonappia sekä laajennuskortti ja korttiin 2 liitettiin yksi painonappi. Kortit olivat yhteydessä toisiinsa I2C-väylän välityksellä, jota kautta ne kommunikoivat keskenään. Laajennuskorttiin liitettiin neljä LED-valoa.

Arduino-korteissa on digitaalisia ja analogisia tuloja. Digitaalisissa tuloissa on mahdollisuus käyttää sisäisiä ylösvetovastuksia, jotka helpottavat kytkimien liittämisessä korttiin, eikä ole tarvetta käyttää ulkoisia ylösvetovastuksia. Tässä tapauksessa kytkintä painettaessa se kytkee kyseisen digitaalisen pinnan maahan (GROUND), jolloin jännite jää ylösvetovastuksen ylitse. Kytkimien toiminta on ohjelmakoodissa asetettu nollalla aktiiviseksi.

Osa Arduino-kortin digitaalisista pinneistä tukee PWM:ää, joten laajennuskortti kytkettiin näihin digitaalisiin pinneihin. Laajennuskortin komponenttien avulla digitaalisen lähdön 5 voltilla ohjataan suurempijännitteistä LED-polttimoa. LED-polttimoiden ohjaus tehdään laajennuskortin avulla.

6 OHJELMISTON TOTEUTUS

6.1 Lohkotason ratkaisu

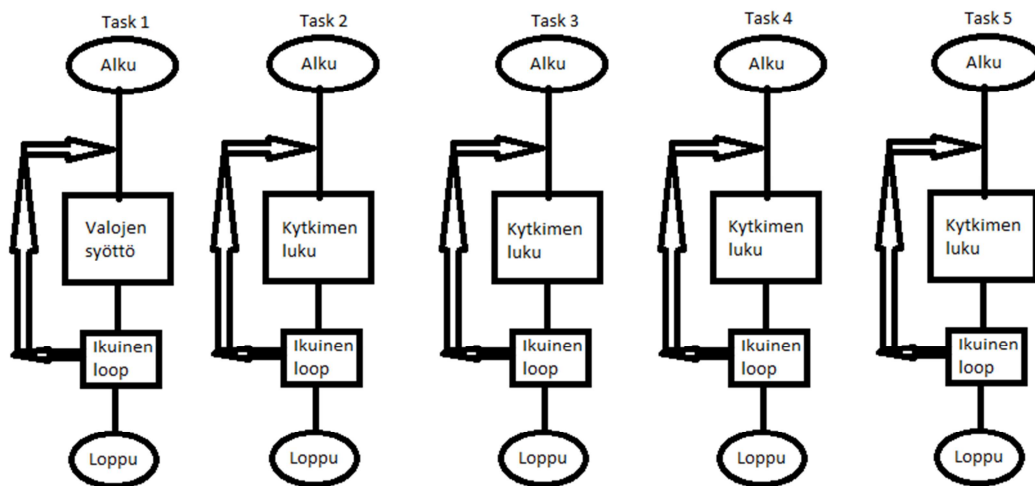
Järjestelmä haluttiin saada toimimaan sillä tavoin, että katkaisijoilla voidaan ohjata yhtäaikaaisesti valojen himmennystä ja himmentymisen reaaliaikaisuutta.

Tämän onnistumiseksi tarvittiin moniajoa. Moniajon suorittaminen olikin hankalampaa, kuin voisi kuvitella, koska Arduino-kortilla täytyi pystyä lukemaan neljää katkaisijaa yhtäaikaaisesti sekä myös ohjata neljää LED-valoa. Tästä syystä päädyttiin käyttämään reaaliaikakäyttöjärjestelmää, jolla moniajo onnistuisi helpommin.

Reaaliaikakäyttöjärjestelmänä käytettiin DuinOS:ää. DuinOS on pieni käyttöjärjestelmä Arduino- ja AVR-pohjaisille korteille. Sen toiminta perustuu preemptiivimoniajoon ja se perustuu FreeRTOS:n kerneliin.

Preemptiivimoniajo on taski, jossa tietokoneen käyttöjärjestelmä käyttää tiettyä kriteeriä päättämään, kuinka pitkään käyttöjärjestelmä suorittaa tiettyä taskia ennen kuin siirtyy suorittamaan seuraavaa taskia. Prosessori siis vaihtelee suoritettavaa taskia esimerkiksi tietyn ajan välein. (6.)

Kuvassa 7 on hahmotelma ohjelmistosta lohkotasolla. Työssä käytetyissä Arduino Uno -korteissa on kolme timeria, joista jokainen voi ohjata kahta PWM-lähtöä, eli käytettävissä olisi kuusi PWM-lähtöä. DuinOS tarvitsee kuitenkin yhtä timeria toimiakseen, joten kaksi PWM-lähtöä jää pois käytöstä. Vain neljä PWM-lähtöä on käytettävissä, kun käytetään DuinOS:ää.



KUVA 7. Rinnakkaisten taskien periaatekuva

6.2 Ohjelmanrakenne

Molempien Arduino-korttien ohjelmiin sisältyy alkumäärittelyt, globaalien muuttujien määrittelyt sekä DuinOS-käyttöjärjestelmä. Lisäksi kortissa 1 on viisi taskia sekä pääohjelma. Kortissa on myös kaksi keskeytystä, joista toinen aktivoituu, kun I2C-väylällä on tulossa dataa, ja toinen, kun väylältä tulee pyyntö lähettää dataa. Kuvasta 7 nähdään kaikki viisi taskia, joita suoritetaan rinnakkain. Kortissa 2 on yksi taski sekä pääohjelma.

Ohjelmakoodin alussa esitellään käytettävät taskit, joiden jälkeen tulee taskifunktiot. Funktioihin tulee taskeissa suoritettavat ohjelmakoodit. Taskifunktioiden jälkeen tulee ohjelman setup-funktio eli asetusfunktio, jossa määritellään esimerkiksi I2C-väylän ja sarjaväylän asetukset, pinnien määritykset tuloiksi tai lähdöiksi sekä luodaan edellä olevat taskit. Taskeille valitaan prioriteetti kolmesta eri tasosta: LOW, NORMAL ja HIGH.

Tässä ohjelmassa on asetettu nappienluku-taskeille LOW-prioriteetti ja valojen syötölle NORMAL.

6.3 Toiminnankuvaus

Arduino-korttien ohjelmointi tehtiin Arduino IDE -sovelluksella, joka on vapaasti ladattavissa valmistajan sivuilta. Arduino-kortit ohjelmoitiin tietokoneella USB-kaapelin välityksellä. Ohjelmaan piti lisäksi hakea DuinOS-

reaaliaikakäyttöjärjestelmäpaketti, jotta se saatiin sisällytettyä projektiin. Ensin ladattu DuinOS:n 0.3-versio ei ollut yhteensopiva Arduino IDE:n 1.0-version kanssa. Lopulta kuitenkin löytyi DuinOS:n 0.4-versio, joka oli yhteensopiva.

Arduino-kortti numero 1 suorittaa viittä taskia rinnakkain sekä lisäksi pääohjelmaa, joka tässä ei käytännössä tee mitään muuta kuin pyörittää viivettä. Ohjelmassa on neljä taskia, joista jokainen seuraa yhden napin painalluksia eli yhteensä neljän napin. Kaikkien toiminto on sama: kun nappia napautetaan, kytkeytyy laajennuskorttiin liitetty valo päälle tai pois. Jos nappia pitää pitkään pohjassa, päällä oleva valo joko kirkastuu tai himmentyy, ja jokaisella uudella pitkään painalluksella suunta muuttuu. Jos valo on pois päältä, kun nappia pidetään pitkään pohjassa, se kytkeytyy päälle ja lähtee kirkastumaan himmeimmästä tasosta. Jokainen katkaisija ohjaa omaa LED-lamppua.

Viides taski syöttää kaikille lampuille muutetun tilan eli päälläolon sekä himmennystason. Kortissa on myös kaksi keskeytystä, joista toinen aktivoituu, kun I2C-väylän isäntä (Arduino-kortti 2) lähettää dataa. Keskeytys vastaanottaa väylältä saapuvan datan ja toimii sen mukaisesti. Toinen keskeytys aktivoituu, kun I2C-väylänisäntä pyytää lähettämään dataa väylälle. Tämän jälkeen kortti lähettää väylälle pyydetyn datan.

Arduino-kortissa 2 on yksi taski sekä pääohjelma, joka myöskään ei tee muuta kuin pyörittää viivettä. Taski lukee korttiin kytketyn napin tilan, ja kun nappi on painettuna, se lähettää dataa I2C-väylän orjalle (Arduino-kortti 2) käskien sammuttaa kaikista LED-lampuista valot.

Arduino-korttien ohjelmakoodit ovat liitteessä 3, mutta ne ovat liikesalaisuuksina saatavissa vain yrityksen sisäiseen käyttöön.

7 TESTAUS

Järjestelmäntestausta tehtiin työn suorituksen aikana jatkuvasti. Laajennuskortin suunnittelun yhteydessä testattiin useita kytkentöjä koekytkentälevyllä, että löydettiin järjestelmään sopiva kokoonpano. Myös laajennuskortin valmistuksen jälkeen suoritettiin kortille testauksia, että juottamisessa ei tullut oikosulkuja eikä myöskään suunnitteluvirheitä. Laajennuskortti oli järjestelmän ensimmäinen protokortti ja se toimi suunnitelman mukaisesti.

Molempien Arduino-korttien ohjelmakoodin tuottamisvaiheessa testattiin lisäominaisuuksia aina ensin erillisellä ohjelmakoodilla. Tällä tavoin saatiin testattua tietyn ominaisuuden toimivuus ennen sen sisällyttämistä varsinaiseen ohjelmakoodiin. PWM:n käyttöönoton yhteydessä ihmeteltiin, kun alun perin oli tarkoitus saada kuusi PWM-lähtöä käyttöön, mutta vain neljä PWM-lähtöä toimi oikein. Syy löytyikin siitä, että DuinOS tarvitsi yhden timerin toimiakseen, joten sitä timeria ei voitu käyttää PWM-lähtöihin.

Järjestelmän toimintaa testattiin myös käyttämällä akkua virtalähteenä. Arduino-kortitkin saivat virtansa akusta, joten USB-kaapelia ei tarvittu tietokoneen ja Arduino-korttien välille virran syöttämiseksi.

8 YHTEENVETO

Insinööriyön tavoitteena oli toteuttaa järjestelmä, jolla voidaan ohjata valaistusta ja himmentää LED-valoja. Työlle asetetut tavoitteet saavutettiin hyvin, eikä tavoitteita jäänyt saavuttamatta. Työn tekeminen oli mielekästä ja tekemisen aikana opin myös paljon uutta Arduino-korteista ja niiden toiminnasta sekä muutenkin AVR-mikrokontrollereista.

Työn tekemisen aikataulutuksen kanssa ei ollut isommin ongelmia, kun kyseessä oli pieni yritys, jossa työtä oli mahdollista tehdä lähes milloin vain. Työtä oli mahdollista tehdä suurimman osan ajasta täyspäiväisesti muiden opintojen päätyttyä, joten se valmistuikin nopealla vauhdilla.

Kokemusta piirilevysuunnittelusta ja ohjelmoinnista on tullut oppitunneilta sekä myös muista projekteista, joissa olen koulun ohella ollut mukana. Työ sopi suuntautumiseeni (elektroniikan suunnittelu ja testaus) sekä muutenkin opintoihini (tietotekniikka). Työtä edeltävistä opinnoista oli hyötyä työn suorittamisen kannalta, kun ei tarvinnut opetella kaikkia alusta. Työtä on tarkoitus jatkokehit-
tää sekä lisätä toiminnallisuuksia.

LÄHTEET

1. Barr, Michael 2007. Introduction to Pulse Width Modulation (PWM). Saatavissa: <http://www.barrgroup.com/Embedded-Systems/How-To/PWM-Pulse-Width-Modulation>. Hakupäivä 15.5.2012.
2. I2C-Bus: What's that?. 2012. Saatavissa: <http://www.i2c-bus.org>. Hakupäivä 16.5.2012.
3. Arduino. 2012. Saatavissa: <http://arduino.cc>. Hakupäivä 16.5.2012.
4. A single master I2C tutorial. 2012. Saatavissa: <http://www.best-microcontroller-projects.com/i2c-tutorial.html>. Hakupäivä 18.5.2012.
5. Rantala, Pekka 2007. I2C-Sarjaväylä. Saatavissa: http://www.oamk.fi/~pekkar/Vaylat/i2c_vayla.pdf. Hakupäivä 18.5.2012.
6. Rouse, Margaret 2005. Preemptive multitasking. Saatavissa: <http://searchcio-midmarket.techtarget.com/definition/preemptive-multitasking>. Hakupäivä 26.5.2012.
7. Arduino Uno. 2012. Saatavissa: <http://arduino.cc/en/Main/ArduinoBoardUno>. Hakupäivä 29.5.2012.

LIITTEET

Liite 1 Lähtötietomuistio

Liite 2 Laajennuskortin suunnitelmat (vain yrityksen sisäiseen käyttöön)

Liite 3 Ohjelmalistaukset (vain yrityksen sisäiseen käyttöön)

OULUN SEUDUN
AMMATIKORKEAKOULU



TEKNIKAN YKSIKKÖ
KOTKANTIE 1, 90250 OULU
www.oamk.fi

LÄHTÖTIETOMUISTIO

Työn tiedot	Tekijä ¹	Tilaaja ²
	Hannila Janne	E. Hannik / Hannila Group
	Tilaajan yhdyshenkilö ja yhteystiedot ³	
	Hannila Esa	
	Työn nimi ⁴	
	Mikroprosessori ohjattu järjestelmä	
Työn kuvaus ⁵		
Valaistuksen ohjaus mikroprosessoreilla, tarviteiden piirilevyjen suunnittelu ja ohjelmointi.		
Työn tavoitteet ⁶		
- Valaistuksen ohjaus - Piirilevyjen suunnittelu - Ohjelmointi		
Tavoiteaikataulu ⁷		
Insinöörityöt valmis 15.6.2012.		
Päiväys ja allekirjoitukset ⁸		
<div> <div>27/2/2012</div> <div>Tekijän allekirjoitus</div> </div> <div> <div>27/2/2012</div> <div>Tilaajan allekirjoitus</div> </div>		
1. Tekijän nimi, puhelinnumero ja sähköpostiosoite. 2. Työn teettävän yrityksen virallinen nimi. 3. Sen henkilön nimi ja yhteystiedot, joka yrityksessä valvoo työn suoritusta. 4. Työn nimi voi olla tässä vaiheessa työnimi, jota myöhemmin tarkennetaan. 5. Työ kuvataan lyhyesti. Siinä esitetään muun muassa työn tausta, lähtötilanne ja työssä ratkaistavat ongelmat. 6. Esitetään lyhyesti ja selvästi työn tavoitteet. 7. Esitetään projektin tavoiteaikataulu. Silloin, kun työllä on välitavoitteita, myös ne merkitään aikatauluun. Tavoiteaikataulun ja oppilaitoksen yleisaikataulun perusteella tekijä laatii oman aikataulunsa. 8. Lähtötietomistio päivätään ja sen allekirjoittavat tekijä ja tilaajan yhdyshenkilö		